IMAGE ACCESS THROUGH METADATA AND CONTENT ANALYSIS IN ARTISTE

(an integrated art analysis and navigation environment)[1]

N Simpkins and P Allen
IT Innovation Centre, 2 Venture Road,
Chilworth Science Park, Southampton, SO16 7NP, UK
Email: artiste@it-innovation.soton.ac.uk

Abstract.

The ARTISTE project will provide an environment for intelligent retrieval and indexing of high resolution images stored in a set of networked databases. This paper focuses on the principals and mechanisms being assessed for use in ARTISTE and in particular the benefits and potentials of the Resource Description Framework being explored in the development of an initial prototype.

1. Introduction

European museums and galleries are rich in cultural treasures but public access has not reached its full potential. Digital multimedia can help address these issues and expand the accessible collections. However, there is a lack of systems and techniques to support both professional and public users in actually accessing such collections. The ARTISTE project [1, 2] is developing a distributed system, which will allow a range of European museums and galleries to have global access to large-scale multimedia repositories.

The ARTISTE system is targeted at two different user communities, museum professionals and publishers. The former includes researchers, art historian and conservators who wish to analyse and compare works of art spread all over the world without leaving their offices. The later includes educational and other image publishers who have been found to have a very diverse set of image access requirements. One of the key requirements is the need for very flexible access to images without advanced knowledge of the indexing concept.

ARTISTE will provide innovative functionality based on automatic content analysis techniques. Their application over a huge collections of high resolution images (20000x20000) will reduce the amount of effort currently needed when dealing with art imagery. Query by visual examples will be supported by the system and the user will be able to search for digitised pictures using colour, texture, shape image analysis techniques in a transparent way.

In addition, the existing distributed collection environments, including legacy data, will be supported by ARTISTE. In the longer term the system can be adapted for public use and use in other image storage domains.

---

[1] ARTISTE brings together four major European galleries (The Uffizi in Florence, The National Gallery and the Victoria and Albert Museum in London and the Louvre related restoration centre, Centre de Recherche et de Restauration des Musées de France) together with NCR, a leading player in database and Data Warehouse technology; Interactive Labs, the new media design and development facility of Italy's leading art publishing group, Giunti; IT Innovation, Web-based system developers; and the Department of Electronics and Computer Science at the University of Southampton. ARTISTE is part funded by the European Commission under the Information Society Technologies Programme (IST).

2. Architecture

The ARTISTE architecture is illustrated in Figure 1. Images of the objects from a collection are held in an object relational database as user defined modules (UDM). These UDMs define both the representations of collection objects and the functions that can be applied to these representations. Functions will include image processing and recognition techniques such as icon and painting technique detection.

Metadata will also be stored in a database. This may be stored in the same object-relational database, or in a separate database, according to the incumbent systems at the user partners.

The distributed query and metadata layer will provide a single interface to the art and its metadata. The metadata layer will also provide facilities to enable queries to be directed towards multiple distributed databases. The end user will be able to seamlessly search the combined art collection. The project builds on the indexing scheme proposed by the AQUARELLE [3] network and will adhere to world-wide digital library standards such as Z39.50, augmenting and extending as necessary to support the richness of metadata standards.

The presentation layer will provide 'simple to use' navigation facilities. Links will be added in real-time to the images retrieved from the database, either through a local copy of the authored links database or through automatic generation from the metadata repository. It will be possible to display the images in a standard Web browser.

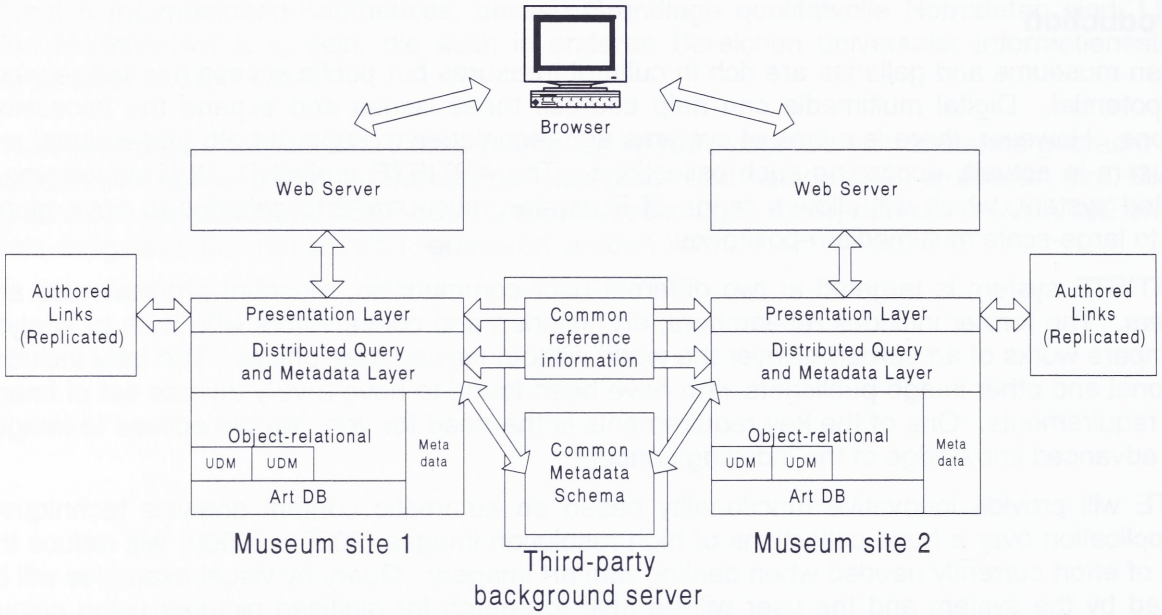

Figure 1 ARTISTE Architecture Outline

3. Indexing, Search and Retrieval

The problem of specifying and maintaining indexes such that all reasonable search queries can be supported has led to a range of more or less complicated solutions. The storage of text documents, for example, has seen the deployment of 'full text indexing engines' which index documents on every content bearing word.

There have been some attempts to provide an image equivalent of full text indexing, by using image 'content analysis'. In general however, it has proven impractical to produce applications with this power. There are very many different aspects of image content that might be used for retrieval, ranging over, for example, colours, patterns, shapes, textures, specific objects, spatial

arrangements or positioning of objects, people etc. Application of algorithms to identify specific features in images is computationally far more expensive than matching index terms.

ARTISTE approaches the indexing and retrieval problem by combining metadata indexing using terms with specialised content analysis algorithms.

Indexing data can be used to directly access images, or to narrow the search for subsequent application of content analysis. Practical response time requirements mean that content analysis may only be applied to a relatively small set of images in 'real time' or to a larger set in batch mode. Depending on the nature and output of a particular algorithm (and user requirements) the results may be stored and become indistinguishable from other static metadata. The distinguishing aspect of such metadata is that for new images it can be automatically generated rather than requiring manual intervention.

ARTISTE is concentrating on a few specific examples of content analysis such as pattern finding, pigment finding and specific shapes. Solutions to these problems will provide tools for specific applications identified in the current user base. Using high-resolution images will allow new comparisons particularly in term of colours. As the state of the art advances the ARTISTE framework will be freely open to the incorporation of new algorithms.

ARTISTE is currently exploring the use of the Resource Description Framework (RDF) and RDF Schemas to describe and encode the metadata in the distributed query layer.

4. Metadata

Museums and galleries typically have a range of metadata associated with their collections. The structure and content of this metadata varies across the organisations. Whilst the scale of existing metadata varies it is not desirable that this must be made consistent or expanded in order for images to be incorporated in the ARTISTE system.

Within ARTISTE different types of data with various different function and origin must be handled:

- the high resolution images themselves
- metadata relating to each image in question (size, format, rights, owner, date produced etc)
- metadata relating to the content of the image in question (subject type, title, date produced etc)
- metadata relating to more than one image. This occurs when for example a single museum object is photographed from more than one angle or in different lighting conditions.

In addition, there are 'external' sources of data and terminology and classification schemes that can benefit ARTISTE. These include at least the multilingual terminology of the NARCISSE [4] project, the Dublin Core [5] and the CIMI [6]. The ARTISTE project would like to reuse these resources, all be it perhaps in some re-coded form.

5. Resource Description Framework

The Resource Description Framework (RDF) [7] provides for metadata definition through the use of schemas. RDF Schemas [8] define the relevant metadata terms (the namespace) and the associated semantics. Individual RDF queries and statements may use multiple schemas. ARTISTE for example, uses existing metadata schemas such as the Dublin Core and will include, by simple URL pointers, other resources (Figure 2).

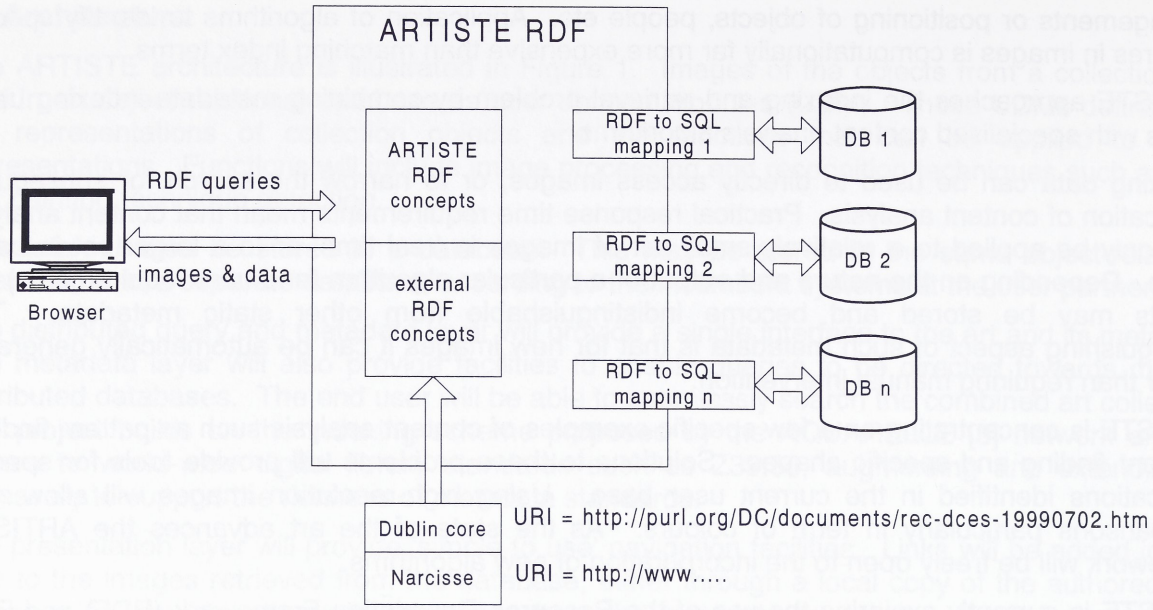

Figure 2 RDF Resources in ARTISTE

RDF provides an XML [9] based mark-up language for describing the relationships between 'resources' in terms of named properties and their values[2]. For example, consider the statement:

"The Mona Lisa was painted by Lenardo da Vinci"

This may be seen as composed of a subject or resource ('The Mona Lisa'), a predication or property ('painted-by'), and an object or literal value ('Lenardo da Vinci'). In RDF, resources are given unique identifiers (URIs). This means that we reference an entity (URI) that represents the resource in question, and then write one, of possibly many, descriptions attributed to that resource. For example:

```
<rdf:RDF>
    <rdf:Description about="http:www.it-innovation.com\art\MonaLisa">
        <artiste:Painter>Lenardo da Vinci</artiste:Painter>
    </rdf:Description>
</rdf:RDF>
```

This entry uses definitions from two 'namespaces' ('rdf' and 'artiste'). The outer enclosing tags label this as an 'RDF' type entry (as defined in the rdf namespace). Internally the entry has a single item, a 'Painter' which has a literal value 'Lenardo da Vinci' which itself is an entry about the resource (URI) on IT Innovation's web site.

We may use specifications from other schemas, such as the Dublin Core within ARTISTE descriptions in a simple fashion. All that is necessary is that the Dublin Core namespace is located and made available to the ARTISTE system. Having evaluated a small number of RDF parsers, we have chosen, currently, the SiRPAC RDF parser [10]. This following the specifications requires that we declare the names spaces we intend to use at the head of an RDF file, for example:

```
<?xml version='1.0'?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
    xmlns:dc=http://purl.org/dc/documents/rec-dces-19990702.htm#
/>
```

[2] We do not intend to provide a comprehensive introduction to RDF here but only a few illustrative examples. The RDF and RDF Schemas are fully described in [7] and [8].

Such a declaration allows us to use the rdf, rdfs (rdf schema) and Dublin Core ('dc') namespaces. The Dublin core for example includes the resource description for 'Title' (of a text, or resource):

```
<rdf:Description ID="Title">
    <rdf:type rdf:resource=
                    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Title</rdfs:label>
    <rdfs:comment>The name given to the resource, usually by the Creator
            or Publisher.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>
```

Using this namespace we might extend the previous description of the Mona Lisa:

```
<rdf:RDF>
    <rdf:Description about="http:www.it-innovation.com\art#MonaLisa">
        <artiste:Painter>Lenardo da Vinci</artiste:Painter>
        <dc:title>Mona Lisa</dc:title>
    </rdf:Description>
</rdf:RDF>
```

Our software can now determine that the title of the '#MonaLisa' resource is 'Mona Lisa' and can also use the comment from that Dublin Core "Title" resource to explain the meaning of this statement (perhaps in different languages, as allowed by the XML attribute 'lang').

The namespaces used in this entry themselves define the relationships and meanings of tags. The means for declaring these new types and attributes are provided by the RDF Schema.

6. ARTISTE RDF Schema

The concepts and their properties that are used in ARTISTE are declared in a schema. The RDF schema mechanism provides a basic type declaration system for use in RDF models. The RDF schema allows us to declare relationships and properties of our own resource types (classes). For example:

```
<rdf:Description rdf:ID=
            "http://it-innovation.soton.ac.uk/artiste/schema/artiste.rdf#Image">
    <rdf:type resource="http://www.w3.org/TR/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/rdf-schema#Resource"/>
    <rdfs:label xml:lang="en">"Image"</rdfs:label>
    <rdfs:comment xml:lang="en">"A stored image together
                    with associated metadata."</rdfs:comment>
</rdf:Description>
```

This declares a new class of object as subclass of 'resource' which has a unique ID (URI) and a label and description. With this we might declare a hierarchy of different image types (images of paintings, images of objects, people etc). At the current time we are working with the generic concept of image and a more extensive range of properties of these (paintings with cracks, etc).

Currently the ARTISTE schema is quite restricted as we have only begun to explore the power of RDF. We are also working within the constraints of an off-the-shelf RDF parser [10], which we have selected after a short evaluation of the available RDF tools but which has not been designed for our requirements.

7. Database Mappings

The concepts represented in the ARTISTE schema are used in the expression of user queries for images and data. In order to execute such a query over the actual collection databases at different sites, a standard SQL query must be produced for each database as each database will have a different logical model (structure) and use different terminology.

The ARTISTE query layer accepts an RDF query and retrieves the relevant images and data by formulating a database specific SQL query for each database that the user has access to. This

process requires that the user query is translated from a generic (database neutral) RDF concept form into SQL which is mapped onto the specific database structures in question. In general, a single RDF formulated query will map onto as many different database (SQL) queries as there are databases active in the ARTISTE network. Not all RDF concepts may be represented in all specific databases, so that for some databases there may be no mapping of an RDF query.

At the top level of the RDF descriptions of databases we define an RDF concept or resource called 'database' which is a sub-class of 'resource'. We then ascribe to a database a number of properties, which include 'mappings'. The ARTISTE schema specifies that a mapping has a number of properties; the RDF concept being mapped and the database (RDF resource) and table and column to which the RDF concept is mapped.

The mapping properties support the translation of an RDF query onto a specific database given the publishing of an RDF set of specific 'mapping' declarations. Each such mapping links an RDF concept with a single database attribute (or table column). There may be multiple mappings for a single concept onto different database attributes (as for example, a table index will commonly be used across several tables).

A trivial mapping of the concept of the 'principal colour' of an image for the database 'IT Innovation collection', for example, has the form:

```
<mapping:DbMapping>
    <mapping:Entity>
        http://intranet.it-
        innovation.soton.ac.uk/artiste/schema/artiste.rdf#PrincipalColour
    </mapping:Entity>

    <mapping:Database>IT innovation collection</mapping:Database>
    <mapping:Table>db2PrincipalColour</mapping:Table>
    <mapping:Column>principalColour</mapping:Column>
</mapping:DbMapping>
```

This maps the concept uniquely identified by the URI 'http://...#PrincipalColour' to 'db2PrincipalColour.principalColour' in the database described by the RDF resource identified as 'IT Innovation collection'. Such mappings are used by the ARTISTE RDF query parser to generate the SQL required for querying any specific TOR database.

8. Future Directions

The current ARTSTE prototype is not intended to be directly developed into the final environment. It is an experimental prototype, exploring the potential of a range of freshly integrated technologies. In particular, we have yet to resolve the problem of integrating Z39.50, with the current RDF concept-based querying architecture. However, we have found a well defined and transparent (with respect to storage issues) platform, which is quite future proof by virtue of a formal structuring and semantics.

We have found that the RDF schema are quite loosely defined, relative to for example a strongly typed programming language. The properties in an RDF schema for example, such as 'number of something', can be specified as being of another type. However, it is clear that we need very strict interpretations of the basic data types (integer, real, percentage etc). This would allow a system to determine, for example, usable interface presentation or input of such data as defined in the RDF schema.

The ARTISTE schema will eventually be registered with a schema registry [11]. It is important that the ARTISTE schema becomes mature and stable before that happens. This is because the publishing of a schema allows others to use it by URI reference. Once published a schema cannot be altered or revoked, at least for some reasonable time or without determining where it has been used, which at this time is not technically feasible.

The ARTISTE project obviously has other fronts for advancement, which include, development of the specific content analysis algorithms and developing a more flexible and powerful GUI.

References

[1] Artiste: An integrated Art Analysis and Navigation Environment, Gert Presutti and Roberto Vaccaro, EVA 27-31 March 2000, Eva2000 Florence.

[2] Artiste web site, http://www.artisteweb.org/, Artiste Consortium, 2000

[3] The Information Network on Cultural Heritage, AQUARELLE, http://aqua.inria.fr/Aquarelle/

[4] Network of Art Research Computer Image Systems in Europe (NARCISSE), http://www.culture.fr/documentation/lrmf/pres.htm, & http://www.1egal2.com/1egal2/presenta/index.htm

[5] Dublin Core Metadata Initiative, http://purl.oclc.org/dc/

[6] Consortium for the Computer Interchange of Museum Information (CIMI), http://www.cimi.org/

[7] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/

[8] Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000, http://www.w3.org/TR/rdf-schema/

[9] Extensible Markup Language (XML) 1.0, W3C Recommendation 10-February-1998, http://www.w3.org/TR/REC-xml

[10] SiRPAC - Simple RDF Parser & Compiler, W3C, http://www.w3.org/RDF/Implementations/SiRPAC/

[11] Forum for Metadata Schema Designers and Implementers, SCHEMAS, http://www.schemas-forum.org/